SAINT LOUIS UNIVERSITY

Computer Vision
Department of computer science
Date:May 6th,2025

Done By:
Ashwin pawar
Bharath Bandlamudi
Puneeth Anantha

Problem Definition

Monocular depth estimation traditionally faces three key challenges: the need for camera intrinsics to obtain metric scale, inability to generalize across domains, and trade-offs between resolution and runtime. Many existing models require specific training datasets or fail to provide depth maps at a resolution and accuracy suitable for practical use cases such as 3D rendering or robotic navigation. The central problem is designing a model that can predict high-resolution, sharp, metric depth maps in a zero-shot setting without additional metadata—efficiently and accurately.

Depth Pro: Sharp Monocular Metric Depth in Less Than a Second

Introduction

Monocular depth estimation—the task of predicting depth from a single RGB image—has witnessed significant advancements in recent years, driven by the growth of deep learning and transformer-based vision models. This problem is central to numerous computer vision applications, such as 3D scene understanding, image editing, augmented reality, and robotics. The goal is to predict a depth map that accurately captures the scene geometry while maintaining metric scale.

Traditional methods often rely on camera intrinsics (e.g., focal length) or domain-specific training data, which limits generalizability. Additionally, many models compromise between speed, resolution, and accuracy. Apple's Depth Pro model addresses these limitations by introducing a foundation model capable of producing sharp, high-resolution, and metrically accurate depth maps from a single image—without requiring any metadata. This documentation explores Depth Pro's architecture, performance, and the practical work done to evaluate its capabilities locally.

Previous Research

Earlier monocular depth estimation methods focused on domain-specific supervised training using RGB-D datasets such as NYU Depth and KITTI. While these approaches could predict metric depth, their performance was constrained to known domains.

MiDaS (Ranftl et al., 2021) shifted the paradigm by enabling zero-shot relative depth estimation through a combination of scale- and shift-invariant losses and diverse data sources. However, MiDaS outputs relative depth, lacking real-world scaling.

Efforts like Metric3D, ZeroDepth, and PatchFusion introduced metric prediction by incorporating camera intrinsics or estimating scene parameters. Nonetheless, these methods struggled with high computational overhead, blurry occlusion boundaries, and reduced accuracy in the absence of metadata. Diffusion-based models such as Marigold improved visual sharpness but suffered from long runtimes.

The need for fast, scalable, and metadata-free metric depth estimation thus remained a largely unmet challenge—until the introduction of Depth Pro.

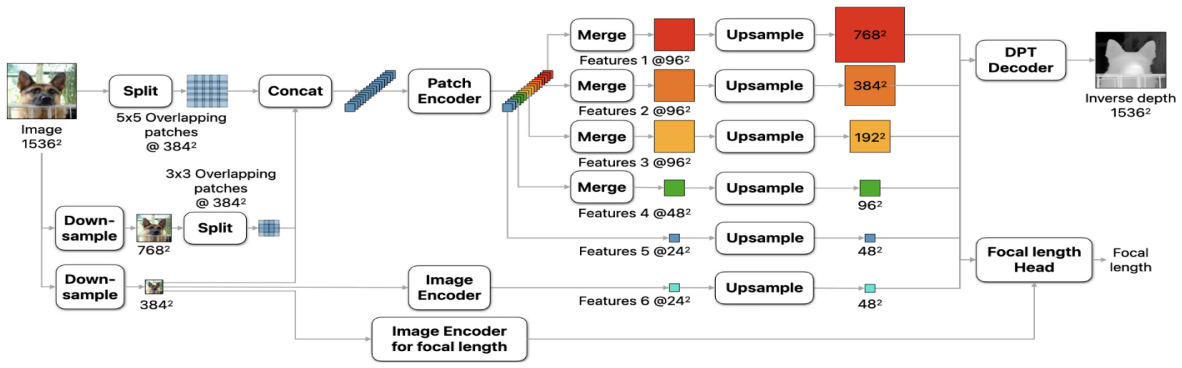
Current Research Approach and Their Architecture in Brief

Depth Pro introduces a novel multi-scale transformer-based architecture that fuses predictions across multiple resolutions to generate sharp and accurate depth maps. The system comprises two main components: a patch encoder and an image encoder, both based on ViT (Vision Transformer) architectures.

The patch encoder processes overlapping image patches at different resolutions, capturing fine-grained spatial details. Simultaneously, the image encoder processes a downsampled version of the entire image, capturing global context. These outputs are fused through a DPT decoder to produce a high-resolution depth map at 1536×1536 resolution.

The training pipeline is divided into two stages: the first trains on a combination of real and synthetic datasets to generalize across domains; the second stage focuses exclusively on synthetic data to refine boundaries and boost sharpness. A separate focal length estimation head predicts the horizontal field of view from image features, eliminating the need for camera metadata.

This design enables Depth Pro to function in real-time (0.3 seconds per image on V100 GPUs) while achieving high accuracy and visual fidelity, even in zero-shot scenarios.



Architecture Diagram

Key Capabilities of Depth Pro

- Predicts high-resolution depth maps (1536×1536) with sharp occlusion boundaries.
- Delivers metric depth estimation in zero-shot settings, without requiring camera intrinsics.
- Uses a multi-scale Vision Transformer architecture for combining global context and fine detail.
- Incorporates zero-shot focal length estimation from intermediate network features.
- Fast inference speed: ~0.3 seconds per image on V100 GPU.

- Competitive performance on standard benchmarks in both accuracy and boundary recall.
- Seamless integration into real-world pipelines using the released pretrained models and code.

Results Section

Depth Pro was extensively benchmarked against state-of-the-art models including DepthAnything v2, Metric3D v2, Marigold, and ZoeDepth. Evaluations spanned across diverse datasets such as Booster, ETH3D, Middlebury, Sun-RGBD, and Sintel.

The model consistently ranked among the top performers in $\delta 1$ score (a key metric measuring percentage of accurate predictions within 25% of ground-truth depth). It also led the field in boundary recall and F1 scores on datasets emphasizing fine-grained occlusion boundaries.

Notably, Depth Pro demonstrated:
- Competitive $\delta 1$ scores across all datasets in zero-shot settings
- State-of-the-art boundary accuracy on both synthetic and real-world datasets
- Leading focal length estimation accuracy across photographic datasets with reliable EXIF data

Its ability to produce sharp depth maps with real-world scale, without needing focal length input or domain-specific tuning, marks a significant leap in the field of monocular depth estimation.

Our Contribution

In our evaluation of the Depth Pro model, we conducted practical work to test its capabilities on local machines. While we did not retrain the model or modify its architecture, we performed the following tasks:

1. Successfully cloned and configured the official Depth Pro GitHub repository.
2. Downloaded the pretrained weights and ran inference scripts on custom images.
3. Implemented a helper script that loads an image, passes it through the Depth Pro model, and visualizes the predicted depth map.
4. Explored strategies for denormalizing the predicted depth output, which is inherently scaled, to map it to real-world units. Though this feature is not yet implemented in the interface But able to give the result in the command line , it represents a future extension of our work.

Our efforts validate the usability of Depth Pro in real-world applications and demonstrate the ease with which pretrained models can be integrated into custom pipelines for visualization and analysis. This foundation sets the stage for future enhancements, including custom training, real-world evaluation, and integration with robotics and 3D reconstruction systems.

Key Resources Required for Implementation

- Pretrained model weights and code available at: https://github.com/apple/ml-depth-pro
- Python environment with PyTorch and TorchVision libraries.
- Access to a CUDA-capable GPU (e.g., NVIDIA V100, RTX 3090) for real-time inference.
- Example image data for testing inference workflows.
- Optionally, datasets with known depth or camera parameters for evaluation or fine-tuning.

Implementation Challenges We Faced

While Depth Pro is designed to be highly accessible via its GitHub repository and pretrained weights, we encountered several practical challenges during local implementation:

1.Environment Setup Issues: Ensuring compatibility between Python versions, PyTorch, and CUDA dependencies required careful handling. Installing the required packages without version conflicts was initially time-consuming.

2. Hardware Constraints: Running the model at its native resolution (1536×1536) requires a GPU with significant VRAM. On some mid-range GPUs, out-of-memory (OOM) errors occurred, requiring us to scale down test images or batch sizes.

3. Model Loading and Inference: While loading the pretrained model was straightforward, adapting the inference scripts to custom images needed some debugging, especially around tensor reshaping and pre-processing.

4. Depth Map Visualization: The output of the model is normalized and not directly human-readable. Designing a helper script to process and visualize this output effectively involved converting tensor data to suitable image formats and applying colormaps.

5. Denormalization Consideration: Although the model produces high-quality normalized depth maps, converting these values back to real-world metric depth involves estimating or predicting the focal length accurately. While this was discussed, we were unable to implement denormalization within the scope of this initial project.

Despite these challenges, we were able to successfully run the Depth Pro model on local machines, generate visually meaningful depth maps, and prepare a framework for future enhancements such as real-time processing and denormalized depth projection.

Drawbacks of the Model

- Struggles with translucent surfaces and volumetric scattering where pixel-level depth is ambiguous.
- Relies on high-quality synthetic datasets for sharpness, which may not capture all real-world variability.
- Current implementation provides normalized output, requiring post-processing for real-world metric scaling.
- May require substantial GPU memory for high-resolution inference, limiting edge-device usability.

Sample Images where model Fails

Graffiti Art – Failure case